(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2019-215794
(P2019-215794A)
(43)公開日　令和1年12月19日(2019.12.19)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/56	(2013.01)	G06F	21/56	370	5J104
H04L 9/10	(2006.01)	H04L	9/00	621A	

審査請求 未請求　請求項の数 12　ＯＬ　（全 14 頁）

(21)出願番号　特願2018-113649 (P2018-113649)
(22)出願日　平成30年6月14日 (2018.6.14)

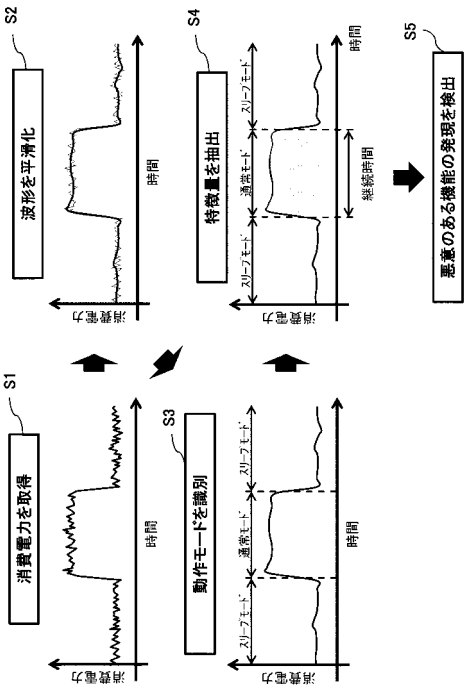
（出願人による申告）平成２９年度、総務省、戦略的情報通信研究開発推進事業（ＳＣＯＰＥ）、重点領域型研究開発、ＩＣＴ重点研究開発分野推進型、「ＩｏＴ部品・機器・ネットワークの階層横断セキュリティ技術の研究開発」委託研究、産業技術力強化法第１９条の適用を受ける特許出願

(71)出願人　899000068
学校法人早稲田大学
東京都新宿区戸塚町１丁目１０４番地
(74)代理人　110002675
特許業務法人ドライト国際特許事務所
(72)発明者　戸川　望
東京都新宿区戸塚町１丁目１０４番地　学校法人早稲田大学内
(72)発明者　長谷川　健人
東京都新宿区戸塚町１丁目１０４番地　学校法人早稲田大学内

Ｆターム(参考)　5J104 AA46

(54)【発明の名称】検出方法及び検出装置

(57)【要約】
【課題】プログラムを書き換え可能な対象デバイスに挿入された悪意のある機能の発現を検出する方法及び装置を提供する。
【解決手段】検出装置の計測器は、対象デバイスの動作中に対象デバイスの消費電力の波形データを取得する（Ｓ１）。検出装置のプロセッサは、計測器より取得された消費電力のデータを平滑化し（Ｓ２）、平滑化後の消費電力のデータをクラスタリングにより分類することによって通常モードとスリープモードとを識別し（Ｓ３）、通常モードと識別された期間ごとに、当該通常モードの継続時間と、当該継続時間での消費エネルギーとを特徴量として抽出し（Ｓ４）、抽出された特徴量から外れ値を求めることで、対象デバイスに挿入された悪意のある機能の発現を検出する（Ｓ５）。
【選択図】図４



【特許請求の範囲】
【請求項１】
プログラムを書き換え可能で且つ複数の動作モードを有する対象デバイスに挿入された機能の発現を検出する方法であって、
前記対象デバイスの動作中に前記対象デバイスのサイドチャネル情報を取得し、
前記サイドチャネル情報から得られるデータをクラスタリングにより分類することによって前記複数の動作モードを識別し、
識別された前記複数の動作モードのうちの少なくとも１つの動作モードで前記対象デバイスが動作している期間について、前記サイドチャネル情報に基づく特徴量を抽出し、
前記特徴量から外れ値を求めることで、前記対象デバイスに挿入された機能の発現を検出する方法。
【請求項２】
前記サイドチャネル情報として、前記対象デバイスの消費電力を取得する、請求項１に記載の方法。
【請求項３】
前記特徴量として、前記少なくとも１つの動作モードの継続時間と、当該継続時間における前記対象デバイスの消費エネルギーと、を抽出する、請求項２に記載の方法。
【請求項４】
前記サイドチャネル情報として、前記対象デバイスに流れる電流を取得する、請求項１に記載の方法。
【請求項５】
前記特徴量として、前記少なくとも１つの動作モードの継続時間と、当該継続時間における前記対象デバイスの累積電流値と、を抽出する、請求項４に記載の方法。
【請求項６】
前記サイドチャネル情報として、前記対象デバイスの抵抗を取得する、請求項１に記載の方法。
【請求項７】
前記特徴量として、前記少なくとも１つの動作モードの継続時間と、当該継続時間における前記対象デバイスの累積抵抗値と、を抽出する、請求項６に記載の方法。
【請求項８】
前記クラスタリングのアルゴリズムとしてＫ平均法を用いる、請求項１～７の何れか１項に記載の方法。
【請求項９】
前記クラスタリングのアルゴリズムとしてウォード法を用いる、請求項１～７の何れか１項に記載の方法。
【請求項１０】
局所外れ値因子法を用いて前記外れ値を求める、請求項１～９の何れか１項に記載の方法。
【請求項１１】
前記サイドチャネル情報から得られるデータを平滑化し、
前記平滑化した後のデータから、前記複数の動作モードを識別する、請求項１～１０の何れか１項に記載の方法。
【請求項１２】
プログラムを書き換え可能で且つ複数の動作モードを有する対象デバイスに挿入された機能の発現を検出する装置であって、
前記対象デバイスの動作中に前記対象デバイスのサイドチャネル情報を取得する計測器と、
前記サイドチャネル情報から得られるデータをクラスタリングにより分類することによって前記複数の動作モードを識別し、識別された前記複数の動作モードのうちの少なくとも１つの動作モードで前記対象デバイスが動作している期間について、前記サイドチャネ

ル情報に基づく特徴量を抽出し、前記特徴量から外れ値を求めることで、前記対象デバイスに挿入された機能の発現を検出するプロセッサと、
を備える検出装置。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、対象デバイスに挿入された機能の発現を検出する方法及び装置に関する。
【背景技術】
【０００２】
近年、プログラムを書き換え可能なプラットフォームが、ＩｏＴ（Internet of Things）デバイス等の様々な電子機器に広く用いられている。ＩｏＴデバイスは、インターネットを介してファームウェアをダウンロードしてアップデートすることができる一方で、情報の漏洩やハードウェアの安全性に関して深刻な懸念が生じている。例えば、インターネットを介してＩｏＴデバイスのファームウェアをアップデートする場合、攻撃者により悪意でファームウェアが改ざんされ、デバイスの機能が容易に変更されてしまうおそれがある。このような事態に対処するため、例えば、非特許文献１には、組み込みシステムにおいて異常なコード実行を検出する技術が開示されている。
【先行技術文献】
【非特許文献】
【０００３】
【非特許文献１】Yannan Liu, Lingxiao Wei, Zhe Zhou, Kehuan Zhang, Wenyuan Xu, and Qiang Xu, “On Code Execution Tracking via Power Side-Channel,” in Proc. ACM SIGSAC Conference on Computer and Communications Security, pp. 1019-1031, 2016.
【発明の概要】
【発明が解決しようとする課題】
【０００４】
このように、プログラムを書き換え可能なプラットフォームを有するデバイスにおいて、攻撃者により悪意でデバイスに挿入された機能（悪意のある機能）の発現を検出する技術が望まれている。
【０００５】
本発明は、プログラムを書き換え可能な対象デバイスに挿入された悪意のある機能の発現を検出する方法及び装置を提供することを目的とする。
【課題を解決するための手段】
【０００６】
本発明に係る検出方法は、プログラムを書き換え可能で且つ複数の動作モードを有する対象デバイスに挿入された機能の発現を検出する方法であって、前記対象デバイスの動作中に前記対象デバイスのサイドチャネル情報を取得し、前記サイドチャネル情報から得られるデータをクラスタリングにより分類することによって前記複数の動作モードを識別し、識別された前記複数の動作モードのうちの少なくとも１つの動作モードで前記対象デバイスが動作している期間について、前記サイドチャネル情報に基づく特徴量を抽出し、前記特徴量から外れ値を求めることで、前記対象デバイスに挿入された機能の発現を検出する。
【０００７】
本発明に係る検出装置は、プログラムを書き換え可能で且つ複数の動作モードを有する対象デバイスに挿入された機能の発現を検出する装置であって、前記対象デバイスの動作中に前記対象デバイスのサイドチャネル情報を取得する計測器と、前記サイドチャネル情報から得られるデータをクラスタリングにより分類することによって前記複数の動作モードを識別し、識別された前記複数の動作モードのうちの少なくとも１つの動作モードで前記対象デバイスが動作している期間について、前記サイドチャネル情報に基づく特徴量を抽出し、前記特徴量から外れ値を求めることで、前記対象デバイスに挿入された機能の発

現を検出するプロセッサと、を備える。
【発明の効果】
【０００８】
本発明によれば、対象デバイスの動作中に取得されるサイドチャネル情報に基づいて、当該対象デバイスに挿入された悪意のある機能の発現を検出することができる。
【図面の簡単な説明】
【０００９】
【図１】本発明の実施形態に係る検出システムの構成を模式的に示すブロック図である。
【図２】対象デバイスの動作モードを説明する模式図である。
【図３】対象デバイスに悪意のある機能が挿入された場合の動作モードを説明する模式図である。
【図４】対象デバイスから悪意のある機能の発現を検出する方法の流れを示す模式図である。
【図５】対象デバイスから悪意のある機能の発現を検出する実験の条件を説明する模式図である。
【図６】対象デバイスの動作中の消費電力の波形を示すグラフである。
【図７】図６の消費電力のデータを平滑化することにより得られる波形を示すグラフである。
【図８】平滑化後の消費電力のデータから、Ｋ平均法によるクラスタリングを用いて通常モードとスリープモードとを識別した結果を表すグラフである。
【図９Ａ】図８の識別結果に基づいて通常モードの期間ごとに得られた、継続時間、消費エネルギー、及び局所外れ値因子の値を示す表である。
【図９Ｂ】図９Ａの実験結果を表すグラフである。
【図１０】ウォード法によるクラスタリングを用いた場合の通常モードとスリープモードとの識別結果を表すグラフである。
【図１１】対象デバイスに流れる電流に基づいて通常モードとスリープモードとを識別した結果を表すグラフである。
【図１２Ａ】図１１に示す識別結果に基づいて通常モードの期間ごとに得られた、継続時間、累積電流値、及び局所外れ値因子の値を示す表である。
【図１２Ｂ】図１２Ａの実験結果を表すグラフである。
【図１３】対象デバイスの動作中の抵抗に基づいて通常モードとスリープモードとを識別した結果を表すグラフである。
【図１４Ａ】図１３に示す実験結果に基づいて通常モードの期間ごとに得られた、継続時間、累積抵抗値、及び局所外れ値因子の値を示す表である。
【図１４Ｂ】図１４Ａの実験結果を表すグラフである。
【発明を実施するための形態】
【００１０】
以下、図面を参照して本発明の実施形態を説明する。
図１は、本発明の実施形態に係る検出システム１の構成を模式的に示すブロック図である。検出システム１は、対象デバイス１０と、電源２０と、検出装置３０と、を備える。検出システム１は、攻撃者により悪意で対象デバイス１０に挿入された機能（悪意のある機能：malfunctions）を検出装置３０により検出するシステムである。
【００１１】
対象デバイス１０は、ＩｏＴ（Internet of Thing）デバイス等の電子機器であり、インターネット等のネットワーク及び／又はユニバーサル・シリアル・バス（ＵＳＢ）等のインターフェースを介してプログラムを書き換え可能なマイクロコントローラを有する。例えば、対象デバイス１０は、ネットワークを介して自己のファームウェアをダウンロードしてアップデートすることができる。対象デバイス１０は、使用時に電源２０に接続され、電源２０からの電源供給を受けて動作する。
【００１２】

対象デバイス１０の動作モードは、通常モード（active mode）とスリープモード（sleep mode）とを含む。通常モードは、対象デバイス１０の通常の処理（主要な機能）を実行するモードである。対象デバイス１０のマイクロコントローラはセンサに接続されており、通常モードにおいてマイクロコントローラは、当該センサから取得された信号をホストコンピュータにネットワークを介して出力する。スリープモードは、対象デバイス１０の必要最小限の機能のみを有効にして消費電力を抑えるモードである。スリープモードでは、対象デバイス１０の主要な機能は停止している。対象デバイス１０の消費電力は時間の経過とともに変化しているが、図２に示すように、通常モードでの消費電力とスリープモードでの消費電力は大きく異なっているため、消費電力によって通常モードとスリープモードとを明確に識別することができる。

【００１３】

対象デバイス１０に挿入され得る悪意のある機能には、対象デバイス１０の動作中は常時アクティブであるタイプと、対象デバイス１０が攻撃者により設定された特定のトリガー条件を満たした時（例えば、タイマーで所定時間経過した時）のみアクティブになるタイプがある。後者のタイプは、悪意のある機能がアクティブにならないとユーザはその存在に気付くことができないため、前者のタイプより厄介である。本実施形態では、後者のタイプの悪意のある機能が対象デバイス１０に挿入されたものとし、図３に示すように、対象デバイス１０が通常モードで動作中に特定のトリガー条件を満たした時のみ、挿入された悪意のある機能がアクティブになるものとする。

【００１４】

図１に戻り、電源２０は、対象デバイス１０が接続されたとき、電源線２０１及びグラウンド線２０２を介して、それぞれ、電源電位及びグラウンド電位を対象デバイス１０に供給する。

【００１５】

検出装置３０は、計測器３１とコンピュータ３２とを備える。計測器３１は、対象デバイス１０の動作中に対象デバイス１０から物理的に外部に漏れる情報（サイドチャネル情報）を取得する装置である。サイドチャネル情報として、消費電力（＝電圧×電流）、電磁波、処理時間等が挙げられるが、本実施形態では、計測器３１により電圧と電流を計測して消費電力を取得するものとする。

【００１６】

計測器３１は、例えば、オシロスコープからなり、プローブ３１２及びプローブ３１４によって計測された対象デバイス１０の電流及び電圧をそれぞれ取得する。具体的に、プローブ３１２は、ケーブル３１１を介して計測器３１に接続され、電源線２０１を流れる電流を計測し、ケーブル３１１を介して計測結果のデータを計測器３１に出力する。プローブ３１４は、ケーブル３１３を介して計測器３１に接続され、電源線２０１とグラウンド線２０２との電位差（電圧）を計測し、ケーブル３１３を介して計測結果のデータを計測器３１に出力する。

【００１７】

計測器３１は、ＣＰＵ（Central Processing Unit）等のプロセッサと、メモリと、ＬＣＤ（Liquid Crystal Display）等の表示部と、を備える。計測器３１のプロセッサは、プローブ３１２及びプローブ３１４からそれぞれ出力された電流及び電圧のデータをサンプリングして消費電力を算出し、時間の経過に伴う消費電力のデータを表示部に表示させるとともに、メモリに格納させる。また、計測器３１はコンピュータ３２に接続されており、算出された消費電力のデータをコンピュータ３２に出力する。

【００１８】

コンピュータ３２は、ＣＰＵ等のプロセッサ３２１と、メモリ３２２と、ＬＣＤ等の表示部３２３と、を備える。プロセッサ３２１は、メモリ３２２に格納されたプログラムにしたがって計測器３１から出力されたデータを解析し、対象デバイス１０に挿入された悪意のある機能の発現を検出する処理を実行する。また、プロセッサ３２１は、処理結果のデータをメモリ３２２に格納させるとともに、表示部３２３に表示させる。コンピュータ

３２により実行される処理については後述する。

【００１９】

なお、表示部を有さない計測器３１を採用し、計測器３１から得られた計測結果をコンピュータ３２の表示部３２３に表示するようにしてもよい。また、図１では、電源２０が対象デバイス１０のみに接続されている例を示しているが、計測器３１及び／又はコンピュータ３２を電源２０と同一の電源系統に接続するようにしてもよい。

【００２０】

次に、図４を参照して、検出装置３０によって実行される、対象デバイス１０に挿入された悪意のある機能の発現を検出する方法の流れについて説明する。

【００２１】

まず、対象デバイス１０の動作中、プローブ３１２及びプローブ３１４を用いて対象デバイス１０の電流及び電圧がそれぞれ計測され、計測器３１のプロセッサは、時間の経過に伴う消費電力（＝電圧×電流）を算出する。これにより消費電力のデータが取得される（ステップＳ１）。取得された消費電力のデータは、コンピュータ３２に出力される。

【００２２】

実際に取得される消費電力のデータはノイズを含んでいるため、コンピュータ３２のプロセッサ３２１は、消費電力のデータを解析する前に消費電力の移動平均をとることにより、当該消費電力のデータを平滑化する（ステップＳ２）。具体的には、時刻ｎにおける消費電力をｘ[ｎ]とすると、Ｎ個の移動平均をとったｙ[ｎ]は式（１）のように表される。

【数１】

$$\mathrm{y}[n]=\frac{1}{N}\sum_{i=0}^{N-1}x[n-i] \quad \cdots(1)$$

【００２３】

次に、プロセッサ３２１は、平滑化後の消費電力のデータをクラスタリングにより２つのクラスに分類し、通常モードとスリープモードとを識別する（ステップＳ３）。ステップＳ３のクラスタリングのアルゴリズムとしてＫ平均法（K-means clustering）を用いることができる。Ｋ平均法は、与えられたデータをその平均値を用いてｋ個のクラスターに分類するものであり、式（２）のように定式化されている。

【数２】

$$\arg\min_{S}\sum_{i=1}^{k}\sum_{x\in s_i}\|x-\mu_i\|^2 \quad \cdots(2)$$

ここで、Ｓ＝｛s_1，s_2，…，s_n｝はクラスターのセットであり、μ_iはクラスターs_iにおける平均値である。本実施形態では２つの動作モード（通常モードとスリープモード）を識別するため、ステップＳ３では２個（ｋ＝２）のクラスターに分類する。なお、本実施形態では、２つの動作モード（通常モードとスリープモード）の消費電力の違いに着目し、消費電力のデータを２個のクラスターに分類しているが、クラスターの数と、識別される動作モードの数とを必ずしも一致させる必要はない。例えば、３個以上のクラスター（ｋ≧３）に基づいて２つの動作モードを識別するようにしてもよい。

【００２４】

なお、ステップＳ３のクラスタリングのアルゴリズムとして、あらかじめ閾値を定め、その閾値を超えるか否かによって通常モードとスリープモードとを識別する方法も考えられるが、閾値をデバイスの種類に応じて調整する必要がある。一方、Ｋ平均法を用いると、閾値を自動的に決定するため、デバイスの種類によらずにクラスタリングを行うことができる。

【００２５】

ステップＳ３の後、プロセッサ３２１は、識別された通常モードとスリープモードのうち通常モードに着目し、通常モードの期間の各々について特徴量を抽出する（ステップＳ４）。ここで、悪意のある機能には、対象デバイス１０の通常動作にいくつかの機能を付加するタイプ（機能付加型：adding-function type）と、対象デバイス１０の通常動作においていくつかの機能を無効にするタイプ（機能無効型：disabling-function type）がある。前者の一例としては内部情報の漏洩があり、後者の一例としてはサービス妨害がある。機能付加型の悪意のある機能が挿入された対象デバイス１０では、その悪意のある機能がない場合に比べて、通常モードにおける消費電力が大きく、且つ継続時間が長い。一方、機能無効型の悪意のある機能が挿入された対象デバイス１０では、その悪意のある機能がない場合に比べて、通常モードにおける消費電力が小さく、且つ継続時間が短い。これらを考慮し、ステップＳ４においてプロセッサ３２１は、通常モードと識別された期間ごとに、通常モードの継続時間と、当該継続時間における消費エネルギー（すなわち、消費電力を継続時間で積分した値）と、を特徴量として算出する。

【００２６】

特徴量を算出した後、プロセッサ３２１は、通常モードの期間ごとに、ステップＳ４で抽出された特徴量から外れ値を求めることで、対象デバイス１０に挿入された悪意のある機能の発現を検出する（ステップＳ５）。ここで、アプリケーションによっては、通常モードの動作においても、継続時間と消費エネルギーがばらつく可能性がある。そこで、本実施形態では、局所外れ値因子（local outlier factor：ＬＯＦ）法を用い、周辺に比べて密度が極端に異なるデータを外れ値とする。プロセッサ３２１は、通常モードの期間ごとに、特徴量（継続時間及びその継続時間における消費エネルギー）からＬＯＦを求め、通常モードと識別された全期間のうちＬＯＦの値が極端に異なる期間に、悪意のある機能が発現したと判断する。

【００２７】

＜実施例＞

次に、対象デバイス１０から悪意のある機能の発現を検出する実施例について、図５～図１４Ｂを参照して説明する。

【００２８】

まず、本実施例の実験条件について説明する。

図５に示すように、対象デバイス１０は、悪意のある機能が発現していないときに以下のi)～iv)の通常動作を行うものとする：i)入力データに対してＡ／Ｄ変換を行う； ii)Ａ／Ｄ変換後のデータに対してＡＥＳ（Advanced Encryption Standard）を用いた暗号化を行う； iii)シリアルインターフェースを介して暗号化されたデータを外部に出力する； iv)次のＡ／Ｄ変換が始まるまでスリープモードで動作する。i)～iii)の動作は通常モードであり、iv)はスリープモードである。i)のＡ／Ｄ変換開始から次のＡ／Ｄ変換開始までの期間は３２ｍｓである。すなわち、対象デバイス１０は、Ａ／Ｄ変換の結果を３２ｍｓごとに暗号化して出力する。対象デバイス１０は、i)～iv)の通常動作を１サイクルとして何サイクルも繰り返すが、本実施例では、５サイクルに１回の割合でＡＥＳ暗号化を無効にする悪意のある機能が発現するものとする。悪意のある機能が発現した場合、Ａ／Ｄ変換の結果が暗号化されずに外部に出力される。

【００２９】

また、電源２０から対象デバイス１０に供給される電圧（Ｖｃｃ）を５．０Ｖ、最大電流を０．４Ａに設定する。

【００３０】

このような条件下で得られた本実施例の実験結果を図６～図９Ｂに示す。

図６は、計測器３１を用いて計測された１秒間の消費電力の波形（Raw data）を示す。図６において横軸は時間[ｓ]を表し、縦軸は消費電力[W]を表す。図６に示すように、スリープモードの消費電力は０．２～０．２２（Ｗ）の間で変動し、通常モードの消費電力は０．２６～０．２８（Ｗ）の間で変動する。

【００３１】

図７に、図６に示された消費電力のデータを平滑化することにより得られた波形を示す。ここで、消費電力の移動平均を算出する際、式（１）においてＮ＝５とした。図７に示すように、スリープモードの消費電力は０．２～０．２１（Ｗ）の間で変動し、通常モードの消費電力は０．２６付近で変動する。このように、消費電力のデータを平滑化することにより、通常モードもスリープモードも変動が小さくなっている。

【００３２】

図８に、平滑化後の消費電力のデータから、Ｋ平均法を用いて通常モード（実線）とスリープモード（破線）とを識別した結果を示す。図９Ａは、識別された通常モードとスリープモードのうち、通常モードの期間ごとに抽出された特徴量（継続時間[ｓ]及び消費エネルギー[ｍＷ・ｓ]）とＬＯＦの値を示す表である。図９Ｂは、図９Ａに示す実験結果を表すグラフである。図９Ｂにおいて、横軸は継続時間[ｓ]を表し、縦軸は消費エネルギー[ｍＷ・ｓ]を表し、背景の陰影の濃さはＬＯＦの値に対応している。陰影が濃いほどＬＯＦの値は低く、陰影が薄いほどＬＯＦの値は高い。図９Ｂの左下の陰影の濃い領域にプロットされたいくつかのデータは、ＬＯＦの値が－１９よりも低く、外れ値として特定される。すなわち、図９Ａの表において、ＬＯＦの値が－１９よりも低い通常モード期間４、９、１４、１９、２４、及び２９において、悪意のある機能が発現したことがわかる。

【００３３】

なお、上述の実施例では、簡単のため、約１秒間で得られた消費電力のデータから悪意のある機能の発現を検出する場合を示したが、言うまでもなく、実際の検出装置３０では、１週間や１か月間等の長期にわたって悪意のある機能の発現を検出する処理を行ってもよい。

【００３４】

本実施形態によれば、対象デバイス１０の通常モードとスリープモードとの消費電力の違いに着目し、通常モードにおける継続時間と消費エネルギーに基づいて悪意のある機能の発現を検出するようにした。これにより、特定のプラットフォームに依存せずに、プログラムを書き換え可能なデバイスに挿入された悪意のある機能の発現を検出することができる。

【００３５】

本発明は、上述の実施形態に限定されるものではなく、本発明の趣旨を逸脱しない範囲内で種々の変更が可能である。

【００３６】

例えば、上述の実施例では、Ｋ平均法を用いて通常モードとスリープモードとを識別したが（図８参照）、クラスタリングのアルゴリズムはＫ平均法に限定されない。例えば、階層的クラスタリングの一つであるウォード法（Ward’s method）を用いてもよい。図１０は、図５に示す実験条件の下で、平滑化後の消費電力のデータからウォード法を用いて通常モード（実線）とスリープモード（破線）とを識別した結果を示している。図１０に示すように、ウォード法を用いた場合においても、通常モードとスリープモードを明確に識別することができる。

【００３７】

他のクラスタリングとして、スペクトラルクラスタリング（spectral clustering）や、Ｋ平均法を派生させたアルゴリズムであるK-means++を適用することも可能である。

【００３８】

さらに、特徴量から外れ値を求めるアルゴリズムはＬＯＦ法に限定されない。例えば、ＤＢＳＣＡＮ（Density-based spatial clustering of applications with noise）を適用することも可能である。

【００３９】

また、上述の方法では、対象デバイス１０のサイドチャネル情報として消費電力を解析することによって、対象デバイス１０に挿入された悪意のある機能の発現を検出したが、他のサイドチャネル情報を解析するようにしてもよい。

【００４０】

図１１、図１２Ａ及び図１２Ｂは、図５に示す実験条件の下で、対象デバイス１０に流れる電流を解析した結果を示す。具体的に、図１１は、平滑化後の電流のデータからクラスタリングによって通常モード（実線）とスリープモード（破線）とを識別した結果を示す。図１２Ａは、図１１に示す識別結果に基づいて通常モードの期間ごとに抽出された特徴量（継続時間とその継続時間での累積電流値）と、特徴量から求められたＬＯＦの値を示す表である。累積電流値は、電流値を継続時間で積分した値である。図１２Ｂは、図１２Ａの実験結果を表すグラフである。図１２Ｂの左下の陰影の濃い領域にプロットされたいくつかのデータは、ＬＯＦの値が－１７よりも低く、外れ値として特定される。すなわち、図１２Ａの表において、ＬＯＦの値が－１７よりも低い通常モード期間４、９、１４、１９、２４、及び２９において、悪意のある機能が発現したことがわかる。

【００４１】

図１３、図１４Ａ及び図１４Ｂは、図５に示す実験条件の下で、対象デバイス１０の動作中の抵抗（＝電圧÷電流）を解析した結果を示す。具体的に、図１３は、平滑化後の抵抗のデータからクラスタリングによって通常モード（実線）とスリープモード（破線）とを識別した結果を示す。図１４Ａは、図１３に示す識別結果に基づいて通常モードの期間ごとに抽出された特徴量（継続時間とその継続時間での累積抵抗値）と、特徴量から求められたＬＯＦの値を示す表である。累積抵抗値は、抵抗値を継続時間で積分した値である。図１４Ｂは、図１４Ａの実験結果を表すグラフである。図１４Ｂの左下の陰影の濃い領域にプロットされたいくつかのデータは、ＬＯＦの値が－２８よりも低く、外れ値として特定される。すなわち、図１４Ａの表において、ＬＯＦの値が－２８よりも低い通常モード期間４、９、１４、１９、２４、及び２９において、悪意のある機能が発現したことがわかる。

【００４２】

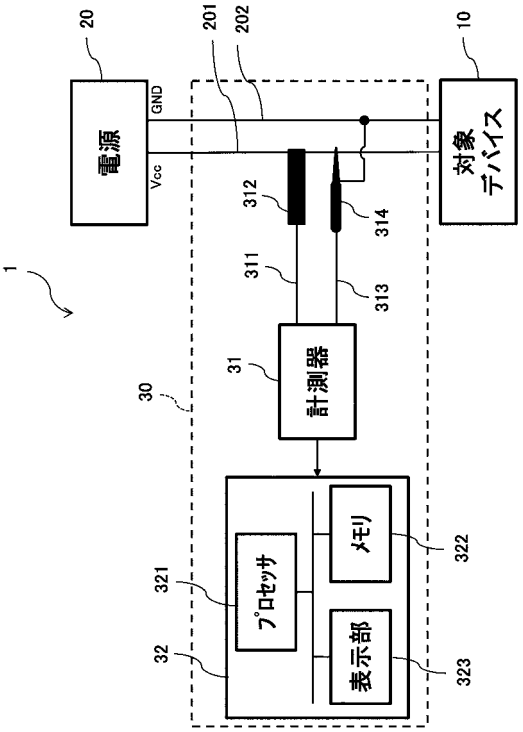
このように、電流、電圧、及び時間を任意に組み合わせた特徴量を用いて、対象デバイス１０に挿入された悪意のある機能の発現を検出することができる。

【符号の説明】

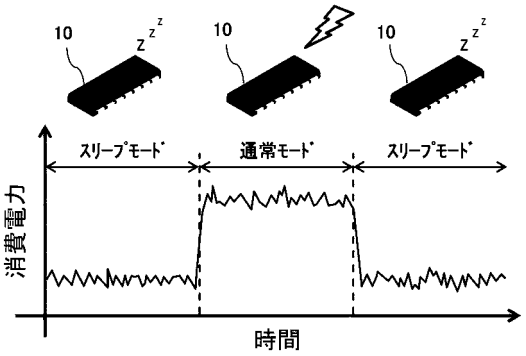
【００４３】

１　検出システム
１０　対象デバイス
２０　電源
３０　検出装置
３１　計測器
３２　コンピュータ
３２１　プロセッサ
３２２　メモリ
３２３　表示部
２０１　電源線
２０２　グラウンド線

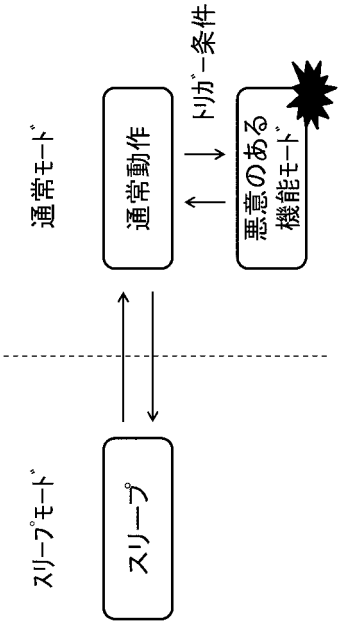
【図１】


20
201
202
10
電源
GND
Vcc
対象
デバイス
312
314
311
313
1
31
計測器
30
321
プロセッサ
メモリ
322
32
表示部
323

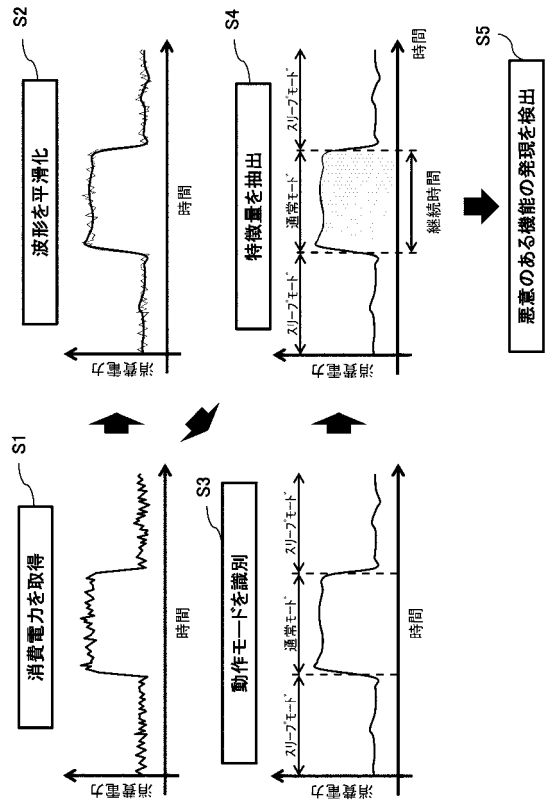
【図２】


10
10
10
スリープモード
通常モード
スリープモード
消費電力
時間

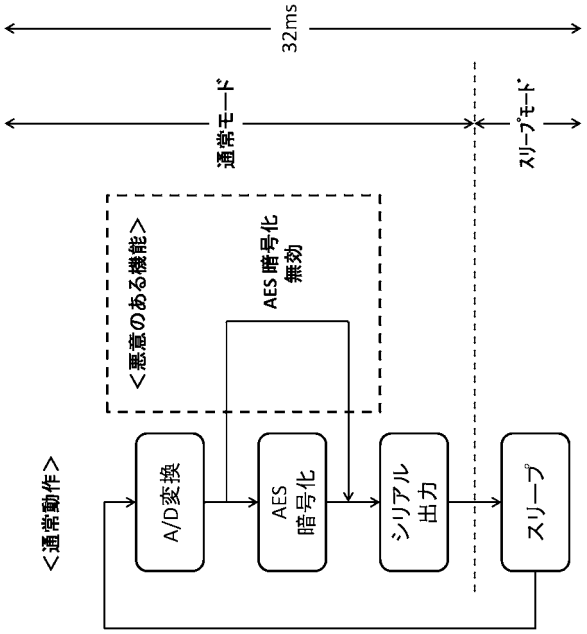
【図３】


通常モード
通常動作
トリガー条件
悪意のある
機能モード
スリープモード
スリープ

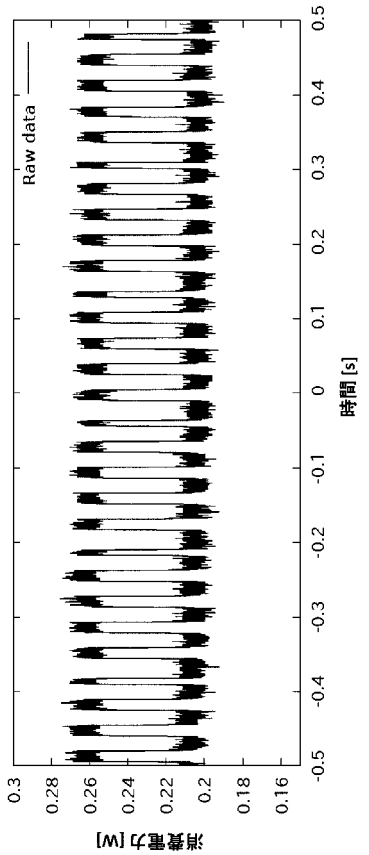
【図４】


S2
波形を平滑化
S4
特徴量を抽出
S5
悪意のある機能の発現を検出
S1
消費電力を取得
S3
動作モードを識別
消費電力
時間
スリープモード
通常モード
継続時間

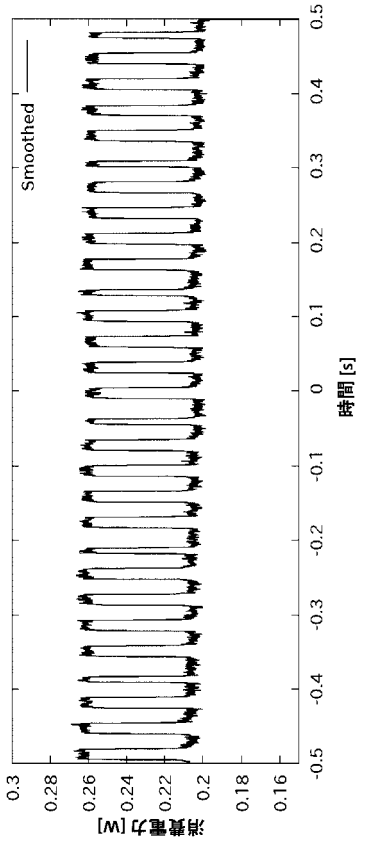
【図５】


32ms
通常モード
スリープモード
<通常動作>
A/D変換
AES
暗号化
シリアル
出力
スリープ
<悪意のある機能>
AES 暗号化
無効

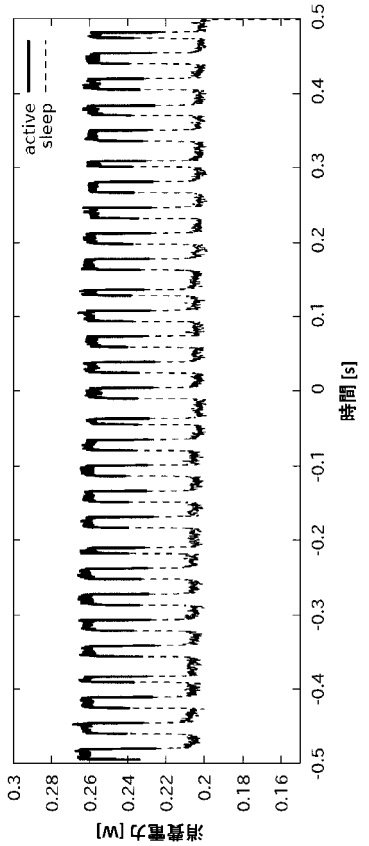
【図６】


Raw data
消費電力 [W]
時間 [s]

【図７】


Smoothed
消費電力 [W]
時間 [s]

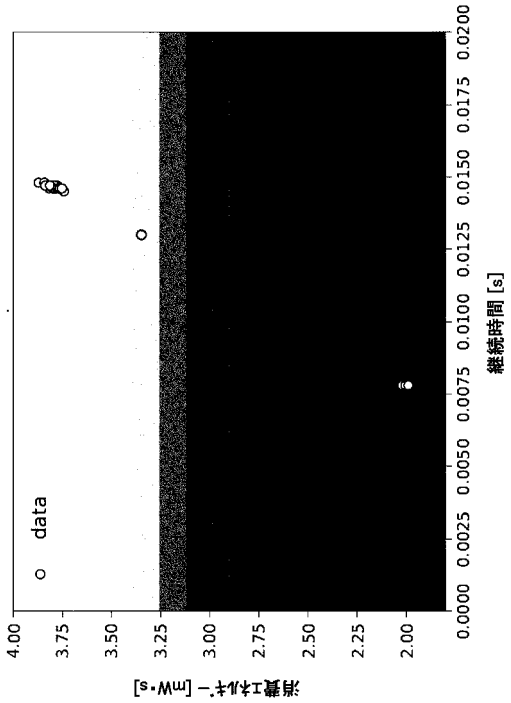
【図８】


active
sleep
消費電力 [W]
時間 [s]

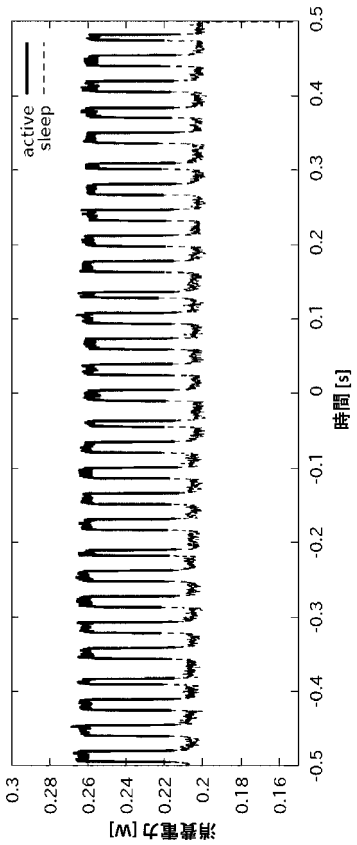
【図９Ａ】

通常ﾓｰﾄﾞ期間	継続時間[s]	消費ｴﾈﾙｷﾞｰ[mW･s]	LOF
1	0.0148	3.8700154	-1.41991882
2	0.0146	3.81662338	-0.9404494
3	0.0147	3.82452424	-1.01141631
4	0.0078	2.0265591	-19.2476636
5	0.0148	3.8412475	-1.14031871
6	0.0147	3.82748164	-1.02368781
7	0.0147	3.82035618	-0.95062244
8	0.0147	3.83693252	-1.10001225
9	0.0078	2.01489426	-19.3259252
10	0.0146	3.7949169	-0.92722028
11	0.0146	3.7910474	-0.93912591
12	0.0146	3.79401588	-0.92722028
13	0.0146	3.78123326	-0.97765742
14	0.0078	2.00408984	-19.3984141
15	0.0147	3.77687538	-1.00122813
16	0.0147	3.79120166	-0.93912591
17	0.0145	3.74185926	-1.12244195
18	0.0146	3.78690232	-0.95683289
19	0.0078	2.02073666	-19.2867275
20	0.0147	3.81237108	-0.9404494
21	0.0146	3.77413986	-1.00122813
22	0.0146	3.7569725	-1.00943257
23	0.0146	3.75492958	-1.01792978
24	0.0078	2.00322512	-19.4042156
25	0.0146	3.76277836	-1.00138305
26	0.013	3.3472639	-8.35758919
27	0.0146	3.7628901	-1.00122813
28	0.0146	3.75163688	-1.04075556
29	0.0078	1.98892634	-19.5001488

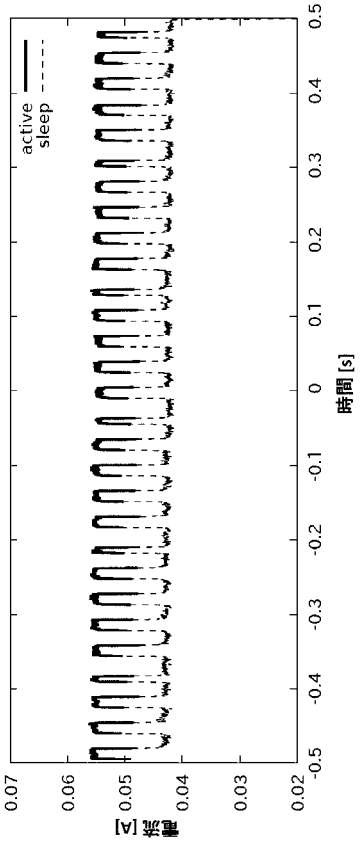
【図９Ｂ】



【図１０】



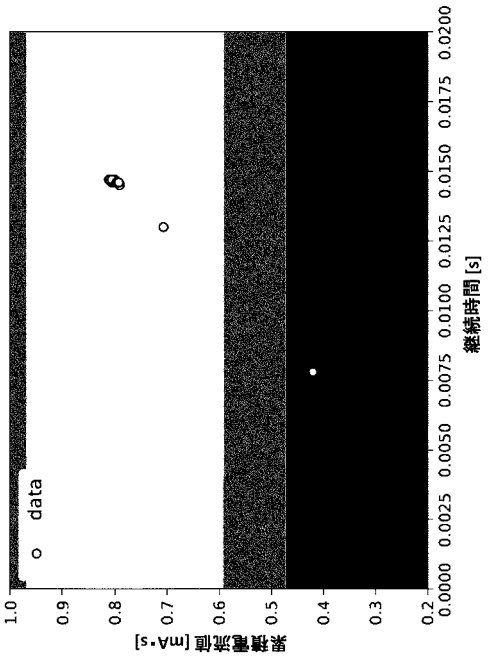
【図１１】



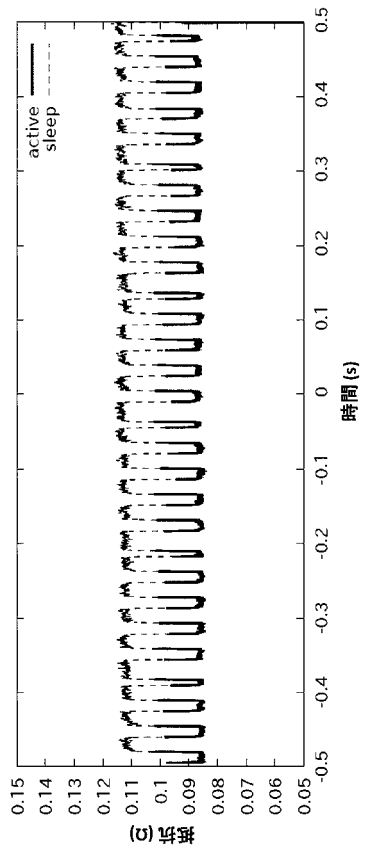
【図１２Ａ】

通常ﾓｰﾄﾞ期間	継続時間 [s]	累積電流値 [mA·s]	LOF
1	0.0147	0.810928	-1.02411
2	0.0146	0.805712	-0.98265
3	0.0147	0.807152	-0.98265
4	0.0078	0.427504	-17.4966
5	0.0147	0.805296	-0.98265
6	0.0146	0.803312	-0.98265
7	0.0147	0.806976	-0.98265
8	0.0147	0.809696	-0.98265
9	0.0078	0.425104	-17.5768
10	0.0146	0.801312	-0.98022
11	0.0146	0.800128	-0.98022
12	0.0146	0.801136	-0.98022
13	0.0146	0.79824	-0.98353
14	0.0078	0.422176	-17.6746
15	0.0146	0.792224	-0.98353
16	0.0147	0.80024	-0.98022
17	0.0145	0.79032	-1.02397
18	0.0146	0.799296	-0.98353
19	0.0078	0.426528	-17.5292
20	0.0147	0.805248	-0.98265
21	0.0146	0.796928	-0.98353
22	0.0146	0.793088	-0.98353
23	0.0146	0.793088	-0.98353
24	0.0078	0.42328	-17.6377
25	0.0146	0.793904	-0.98353
26	0.013	0.706464	-6.32314
27	0.0146	0.794496	-0.98353
28	0.0146	0.791808	-0.98353
29	0.0078	0.42	-17.7473

【図１２Ｂ】



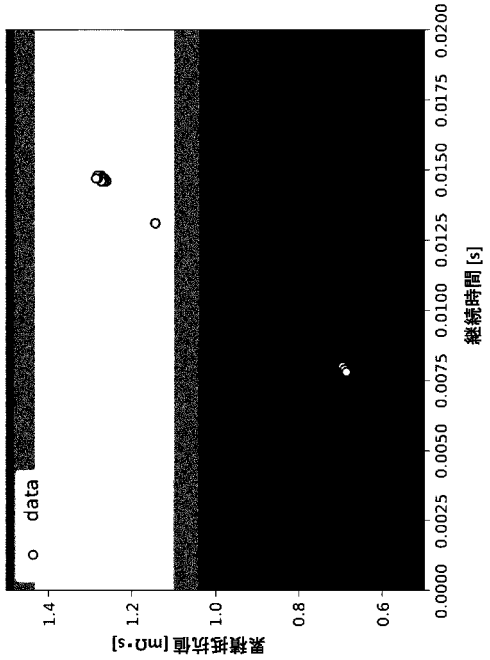
【図１３】



【図１４Ａ】

通常ﾓｰﾄﾞ期間	継続時間 [s]	累積抵抗値[mΩ·s]	LOF
1	0.0148	1.274738	-0.92737
2	0.0148	1.273616	-0.92737
3	0.0148	1.279243	-0.97627
4	0.0079	0.685064	-28.6354
5	0.0148	1.28276	-1.03159
6	0.0147	1.267186	-1.0346
7	0.0147	1.268625	-1.02542
8	0.0147	1.265553	-1.05562
9	0.0079	0.688829	-28.5228
10	0.0147	1.270446	-0.96843
11	0.0148	1.282573	-1.02308
12	0.0146	1.260763	-1.28072
13	0.0146	1.265657	-1.05233
14	0.0079	0.694574	-28.3509
15	0.0147	1.28542	-1.15244
16	0.0147	1.280141	-0.97627
17	0.0147	1.279828	-0.97627
18	0.0147	1.274084	-0.92737
19	0.008	0.696044	-28.3069
20	0.0147	1.271319	-0.96843
21	0.0146	1.267488	-1.03268
22	0.0146	1.273965	-0.92737
23	0.0146	1.273178	-0.94452
24	0.0079	0.690666	-28.4678
25	0.0146	1.273242	-0.94452
26	0.0131	1.144143	-11.3384
27	0.0147	1.281272	-0.98887
28	0.0147	1.286039	-1.18057
29	0.0078	0.686607	-28.5893

【図１４Ｂ】


data
累積抵抗値 [mΩ·s]
継続時間 [s]
1.4
1.2
1.0
0.8
0.6
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200